

delete:

操作符用于删除对象的某个属性

```
1 var obj = {a:10, b:200, c: 500};
2     delete obj.b;//删除属性
3     //obj.b = null;  //只是把属性值改为了null，实际这个属性还是存在的
4     console.log(obj);
5
6 //如果你使用delete操作符删除了一个属性，再次用in检查时，会返回false，如：
7 var mycar = {
8     make: 'Honda',
9     model: 'Accord',
10    year: 1998};
11 delete mycar.make;
12 console.log('make' in mycar); //false
13
14 //如果你把一个属性值设为undefined，但是没有使用delete操作符，使用in检查，会返回true.
15 var mycar = {
16     make: 'Honda',
17     model: 'Accord',
18     year: 1998};
19 mycar.make = undefined;
20 console.log('make' in mycar); //true
```

三目运算:

语法结构：**表达式1 ? 表达式2 : 表达式3**

执行思路:

如果表达式1为true，则返回表达式2的值,如果表达式1为false，则返回表达式3的值

```
1 //三目运算符用的很多
2 var gender = 0;
3     console.log(gender == 1 ? '帅哥' : '美女');//美女
4     //语法没有错误，但是业务逻辑不严谨
5
6 //三目运算符是可以嵌套的
```

```

7 var g = gender == 1 ? '帅哥' : (gender == 2 ? '美女' : '保密');
8     console.log(g); // 保密
9
10 // 三目运算符
11 var gender = 0;
12     console.log(gender == 1 ? 'A' : 'B'); // B
13 var g = gender == 1 ? '时光' : (gender == 2 ? '技术' : '课件');
14     console.log(g); // 课件

```

逗号运算符:

```

1 var gender = (0, 2, 3, 500, b = 200, false, new Date(), true);
2     console.log(gender); // true 返回最后一个值

```

eval():

eval()方法就是一个js语言的执行器，它能把其中的参数按照JavaScript语法进行解析并执行

```

1 eval('var b = 300;');
2     console.log(b); // 300
3
4 eval('var r = 100*25 - 300/4; console.log(r);'); // 2425

```

eval()方法的参数形式

如果参数不是字符串，而是整数或者是Function类型，则直接返回该整数或Function.

```

1 var x = 1;
2 var y = eval(x);
3     console.log(y); // eval()方法返回1
4 var z = eval(1);
5     console.log(z); // eval()方法返回1

```

如果参数是字符串，并且字符串中是表达式，则eval()方法会计算字符串中的表达式，返回该表达式计算的结果.

```

1 eval(" 'true' "); // 返回字符串"true"
2 eval(" true "); // 返回Boolean类型的true
3 eval(" 1 "); // 返回整数1

```

```
4 eval(" [0,1,2] ");//返回数组[0,1,2]
5 eval(" 2 + 2 ");//返回整数4
6 eval(" 2 + '2' ");//返回字符串"22"
```

如果参数是字符串，并且字符串中的表达式是JSON格式（此时的JSON格式外面要加上一对小括号将JSON包围起来），则eval()方法返回该JSON。

```
1 var j = '{"aa':11,'bb':22}";
2 eval("(" + j + ")");
3 //返回{aa: 11, bb: 22}
4
5
6 eval("({aa':11,'bb':22})");
7 //返回{aa:11,bb:22}
8
9
10 eval("({aa:11,bb:22})");
11 //js中{aa:11,bb:22}这种格式在解析时会转换成{'aa':11,'bb':22}这种格式
12
13
14 eval("{'aa':11,'bb':22}");
15 //抛出异常
```

循环：

(2条消息) js中 switch 语句用法详解和 for in、for of循环相关用法_陈坚泓的博客-CSDN博客_js switch循环

switch循环:

```
1 switch(表达式){
2     case value1:
3         //表达式等于 value1 时要执行的代码
4         break;
5     case value2:
6         //表达式等于value2 时要执行的代码
7         break;
8     default:
9         //表达式不等于任何一个value时要执行的代码
```

- switch：开关转换，case：小例子选项
- 关键字 switch 后面括号内可以是表达式或值，通常是一个变量
- 关键字 case，后跟一个选项的表达式或值，后面跟一个冒号
- switch 表达式的值会与结构中的 case 的值做比较
- 如果存在匹配全等(==)，则与该 case 关联的代码块会被执行，并在遇到 break 时停止，整个 switch 语句代码执行结束
- 如果所有的 case 的值都和表达式的值不匹配，则执行 default 里的代码
- 执行case 里面的语句时，如果没有break，则继续执行下一个case里面的语句

```
1  var week = 2;
2      switch (week) {
3          case 0:
4              console.log('星期天');
5              break;
6          case 1:
7              console.log('星期1');
8              break;
9          case 2:
10             console.log('星期2');
11             break;
12          case 3:
13             console.log('星期3');
14             break;
15          case 4:
16             console.log('星期4');
17             break;
18          case 5:
19             console.log('星期5');
20             break;
21
22          case 6:
23             console.log('星期6');
24             break;
25          default:
26             console.log('请输入正确的数字');
27             break;
28      }
```

```

29
30 //判断天数
31 var year = 2000, month = 2;
32     switch (month) {
33         case 1: case 3: case 5:
34             case 7: case 8: case 10: case 12:
35                 console.log('31天');
36                 break;
37             case 4: case 6: case 9: case 11:
38                 console.log('30天');
39                 break;
40             case 2:
41                 if ((year % 4 === 0 && year % 100 !== 0) || year % 400 === 0) {
42                     console.log('29天');
43                 } else {
44                     console.log('28天');
45                 }
46                 break;
47             default:
48                 console.log('请输入正确的数字');
49                 break;
50     }

```

for 循环:

```

1  for(初始化变量;条件表达式;操作表达式)
2  {
3      //循环体
4  }

```

```

1  for (var i = 0; i < 10; i += 2) { console.log(i); }
2  for (var j = 10; j > -3; j--) { console.log(j); }
3
4  //如果循环体只有一行代码，{}是可以省略的
5  for (var n = 1; n < 20; n *= 3) console.log(n);
6
7  //初始化语句或者条件修改语句可以放在()之外

```

```
8  var m = 20;
9  for (; m > 3;) {
10     console.log(m);
11     m /= 5;
12 }
13
14 //循环的嵌套
15 for (var x = 1; x <= 9; x++) {
16     var arr = [];
17     for (var y = 1; y <= x; y++) {
18         arr.push(`${y}x${x}=${x * y}`);
19     }
20     console.log(arr.join(', '));
21 }
22
23 //100钱100鸡  公鸡 5文钱  母鸡3文钱  小鸡 1文钱3个
24 for (var gj = 0; gj < 20; gj++) {
25     for (var mj = 0; mj < 33; mj++) {
26         if (gj * 5 + mj * 3 + (100 - gj - mj) / 3 == 100) {
27             console.log(gj, mj, 100 - gj - mj);
28         }
29     }
30 }
```